

CLAIMS

I/We claim:

1. A method for parsing a markup file, comprising:
- 5 parsing a first portion of the markup file with a lightweight parser in a computer system, the lightweight parser being capable of performing a first set of parsing tasks;
- parsing a second portion of the markup file with a heavyweight parser in the computer system, the heavyweight parser being capable of
- 10 performing a second set of parsing tasks, wherein the first set of parsing tasks is a subset of the second set of parsing tasks; and
- transitioning between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser upon an occurrence of a
- 15 transition event.
2. The method of claim 1, further comprising detecting an occurrence of the transition event comprising a requirement that the lightweight parser perform a parsing task excluded from the first set of parsing
- 20 tasks.
3. The method of claim 1, wherein the step of parsing a first portion of the markup file with a lightweight parser further comprises establishing a channel applying the first portion of the markup file to the lightweight parser and directing a number of events generated by the lightweight parser to an
- 25 application.
4. The method of claim 1, wherein the step of parsing a second portion of the markup file with a heavyweight parser further comprises
- 30 establishing a channel to apply the second portion of the markup file to the heavyweight parser and to direct events generated by the heavyweight parser to an application.

5. The method of claim 1, further comprising maintaining an events stack in the computer system, the events stack having a number of open events from the lightweight parser.

5 6. The method of claim 5, wherein the step of maintaining an events stack in the computer system, further comprising:

storing a number of open events from the lightweight parser in the events stack;

10 matching a closing event with one of the opening events in the events stack; and

deleting the opening event matching the closing event from the event stack.

15 7. The method of claim 5, wherein the step of transitioning between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser further comprises:

generating a priming file from the events stored in the events stack; and

20 applying the priming file to the heavyweight parser.

25 8. The method of claim 7, wherein the step of transitioning between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser further comprises discarding a number of priming events generated by the heavyweight parser upon the application of the priming file thereto.

30 9. A program embodied in a computer readable medium employed to parse a markup file, comprising:

a lightweight parser that performs a first set of parsing tasks to parse the markup file;

code that transitions a parsing of the markup file from the lightweight parser to the heavyweight parser upon an occurrence of a transition event.

11. The program embodied in the computer readable medium of claim 9, further comprising code that establishes a channel that applies the first portion of the markup file to the lightweight parser and that directs a number of events generated by the lightweight parser to an application.

13. The program embodied in the computer readable medium of claim 9, further comprising code that maintains an events stack in a memory, wherein a number of open events from the lightweight parser are stored in the events stack.

14. The program embodied in the computer readable medium of claim 13, wherein the code that maintains the events stack in the memory, further comprises:

code that stores a number of open events from the lightweight parser in the events stack;

code that matches a closing event with one of the opening events in the events stack; and

code that deletes the opening event matching the closing event from the event stack.

5

15. The program embodied in the computer readable medium of claim 13, wherein the code that transitions the parsing of the markup file from the lightweight parser to the heavyweight parser further comprises:

code that generates a priming file from the events stored in the events stack; and

code that applies the priming file to the heavyweight parser.

16. The program embodied in the computer readable medium of claim 15, wherein the code that transitions the parsing of the markup file from the lightweight parser to the heavyweight parser further comprises code that discards a number of priming events generated by the heavyweight parser upon the application of the priming file thereto.

17. A system for parsing a markup file, comprising:
first means for parsing the markup file employing a first set of parsing tasks;

second means for parsing the markup file employing a second set of parsing tasks, wherein the first set of parsing tasks is a subset of the second set of parsing tasks; and

means for transitioning a parsing of the markup file from the first means to the second means upon an occurrence of a transition event.

18. The system of claim 17, wherein the transition event further comprises a requirement that the first means perform a parsing task excluded from the first set of parsing tasks.

19. The system of claim 17, further comprising means for establishing a channel applying the first portion of the markup file to the lightweight parser and directing a number of events generated by the lightweight parser to an application.

5

20. The system of claim 17, further comprising means for establishing a channel applying the second portion of the markup file to the heavyweight parser and directing events generated by the heavyweight parser to an application.

10

21. A method for parsing a markup file, comprising:

- parsing the markup file with a lightweight parser in a computer system, the lightweight parser being capable of performing a first set of parsing tasks;
- maintaining an events stack in the computer system by storing a number of open events generated by the lightweight parser in the events stack and deleting select ones of the open events previously stored in the events stack that match corresponding ones of a number of closing events generated by the lightweight parser;
- detecting an occurrence of a transition event in the computer system, the transition event comprising a requirement that the lightweight parser perform a parsing task excluded from the first set of parsing tasks; and
- transitioning from the parsing of the markup file with the lightweight parser to parsing the markup file with the heavyweight parser in the computer system upon an occurrence of the transition event by:
 - ceasing an application of the markup file to the lightweight parser;
 - generating a priming file from the current ones of the events stored in the events stack and applying the priming file to the heavyweight parser; and
 - discarding a number of priming events generated by the heavyweight parser upon the application of the priming file thereto; and

15

20

25

30

Approved for Release

parsing a remaining portion of the markup file with the heavyweight parser after the application of the priming file, the heavyweight parser being capable of performing a second set of parsing tasks, wherein the first set of parsing tasks is a subset of the second set of parsing tasks.

5

22. A system for parsing a markup file, comprising:

a processor circuit having a processor and a memory;

a hybrid parser stored in the memory and executable by the

10 processor, the hybrid parser comprising:

a lightweight parser that performs a first set of parsing tasks to parse the markup file;

a heavyweight parser that performs a second set of parsing tasks to parse the markup file, wherein the first set of parsing tasks is a subset of the second set of parsing tasks;

15

an events stack stored in the memory;

logic that maintains the events stack by storing a number of open events generated by the lightweight parser in the events stack and deleting select ones of the open events previously stored in the events stack that match corresponding ones of a number of closing events generated by the lightweight parser; and

20

logic that transitions a parsing of the markup file from the lightweight parser to the heavyweight parser upon an occurrence of a transition event comprising:

25

logic that ceases applying the markup file to the lightweight parser;

logic that generates a priming file from the current ones of the events stored in the events stack and applies the priming file to the heavyweight parser;

30

2025-07-01 10:09:00

logic that discards a number of priming events generated by the heavyweight parser upon the application of the priming file thereto; and

5

logic that applies a remaining portion of the markup file to the heavyweight parser after the application of the priming file.

T05020"01051000